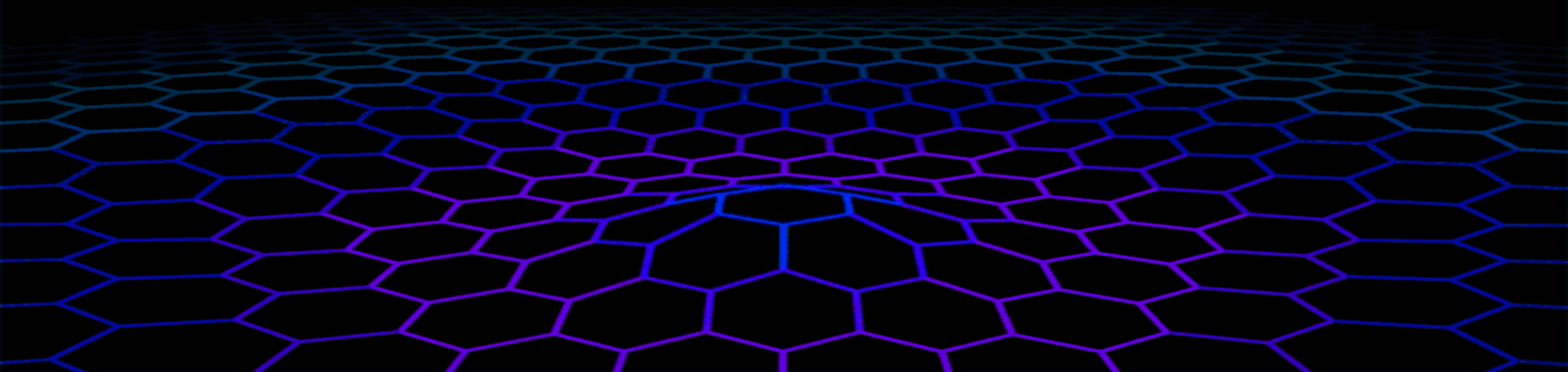


Sistemas Operacionais



Sistemas Operacionais

Sistema de Numeração

Conjunto de símbolos alfanuméricos adotados na representação de quantidades

Estabelecimento de regras que regem a forma de representação

Cada sistema de numeração

Forma diferente de representação de quantidades

Inalteração das quantidades

Alteração *apenas* dos símbolos usados para representá-las

Bases e Representações Numéricas

Base

Quantidade igual ao número de algarismos que compõem um sistema de numeração

Representações

Posicional

Não-posicional

Sistemas de Numeração Não Posicionais

Valor atribuído a um símbolo é inalterável, independente da posição em que se encontre no conjunto de símbolos que representam uma quantidade.

Sistema de Numeração Romano

<i>X</i>	<i>X</i>	<i>I</i>
<u>10</u>	<u>10</u>	<u>1</u>

<i>X</i>	<i>I</i>	<i>X</i>
<u>10</u>	<u>1</u>	<u>10</u>

Sistemas de Numeração Posicionais

Valor atribuído a um símbolo dependente da posição em que se encontre no conjunto de símbolos que represente uma quantidade.

Sistema de Numeração Decimal

5	7	3
500	70	3

3	5	7
300	50	7

Sistema de Numeração

Sistemas de Numeração Típicos em Interações Usuário-Computador

Decimal

Binário

Octal

Hexadecimal

Exemplos de Sistemas de Numeração

Sistema	Base	Algarismos
Binário	2	<i>0, 1</i>
Ternário	3	<i>0, 1, 2</i>
Octal	8	<i>0, 1, 2, 3, 4, 5, 6, 7</i>
Decimal	10	<i>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</i>
Duodecimal	12	<i>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B</i>
Hexadecimal	16	<i>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F</i>

Padrões de Representação

Padrões de Representação

Letra após *número* para indicação da *base*

Número entre *parênteses* e *base* como *índice do número*

Número e *base* como *índice do número*

Exemplo - Sistema *Decimal*

3964_D

(3964)₁₀

3964₁₀

Sistema Decimal - Base 10

Sistema mais utilizado por seres humanos

Uso de *10* símbolos para a representação de quantidades



Peso

Ponderação em função da posição do algarismo no número

Potências da base em função da *unidade* ($1 = 10^0$)

10^1 unidades - *dezena*

Sistema Decimal - Base 10

Exemplo - 3964

4 unidades, 6 dezenas, 9 centenas e 3 milhares

$$3000 + 900 + 60 + 4 = 3964$$

3	9	6	4
3000	900	60	4
<hr/>			<hr/>
3×10^3	9×10^2	6×10^1	4×10^0

Sistema Decimal - Base 10

Montando um número na base decimal

$$\begin{aligned}\text{Ex.: } 328451,52_{10} &= 3 \times 10^5 + 2 \times 10^4 + 8 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} \\ &\quad + 2 \times 10^{-2} \\ &= 300000 + 20000 + 8000 + 400 + 50 + 1 + 0,5 + 0,02 \\ &= 328451,52_{10}\end{aligned}$$

Sistema Decimal - Base 10

Numeração **decimal** (base **10**)

Símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Característica de valor posicional (casa)

Unidades (1s), dezenas (10s), centenas (100s), milhar (1000s),...

Exemplo:

Número **736**

$$6 \times 1 = 6$$

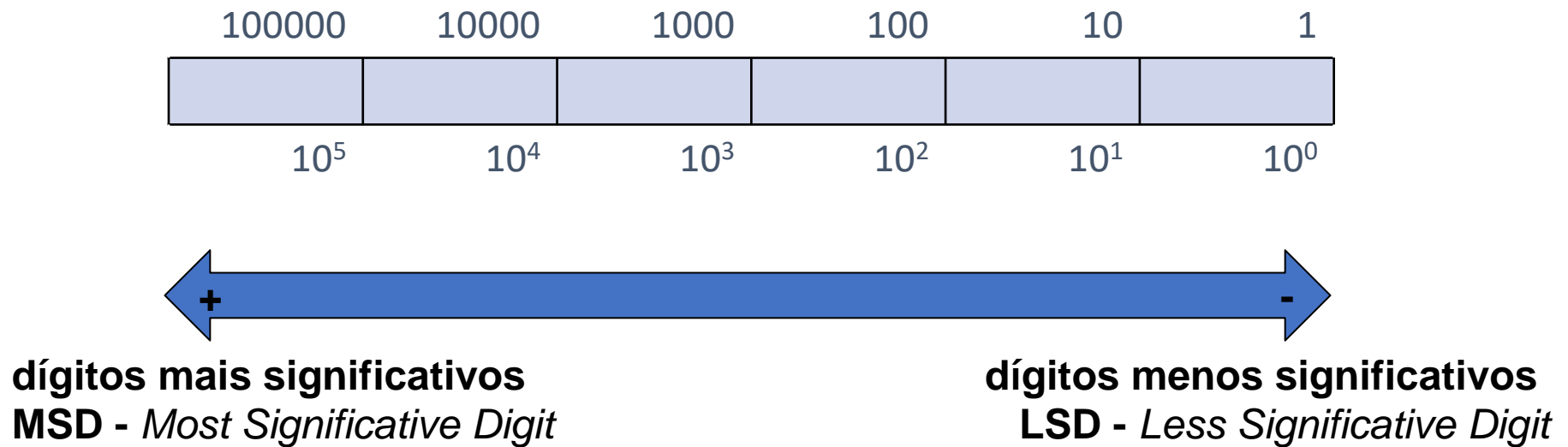
$$3 \times 10 = 30$$

$$7 \times 100 = 700$$

$$6 + 30 + 700 = 736$$

Sistema Decimal - Base 10

Posições:



Sistema Decimal - Base 10

Exemplo:

100000	10000	1000	100	10	1
0	0	2	5	8	0
10^5	10^4	10^3	10^2	10^1	10^0

**O número “dois mil quinhentos e oitenta” decimal é obtido:
 $(2 \times 1000) + (5 \times 100) + (8 \times 10) = 2000 + 500 + 80 = 2580$**

Sistema Binário - Base 2

Sistema Binário de numeração é o principal sistema dos PCs;

Este sistema de numeração, como o próprio nome sugere, apresenta base 2. Os números 0 e 1 são os dígitos deste sistema;

O sistema binário é de grande importância, pois apresenta correspondência direta com os estados de um sistema digital.

Por exemplo:

Para o dígito 0 pode-se atribuir o valor desligado e para o dígito 1 pode-se atribuir o valor de ligado.

Sistema Binário - Base 2

Uso de 2 símbolos para a representação de quantidades



Validade dos conceitos de *peso* e *posição*

Posições *não* recebem denominações específicas (como no sistema decimal)

Denominação genérica de cada algarismo - *Bit* (*Binary digit*)

Sistema Binário - Base 2

Destaque para os algarismos extremos dos números

Algarismo mais à esquerda - *Most Significant Bit* (MSB)

Algarismo mais à direita - *Less Significant Bit* (LSB)

1 **0** **1** **1**
MSB | | **LSB**

Sistema Binário - Base 2

Numeração **binária** (base **2**)

Símbolos 0, 1

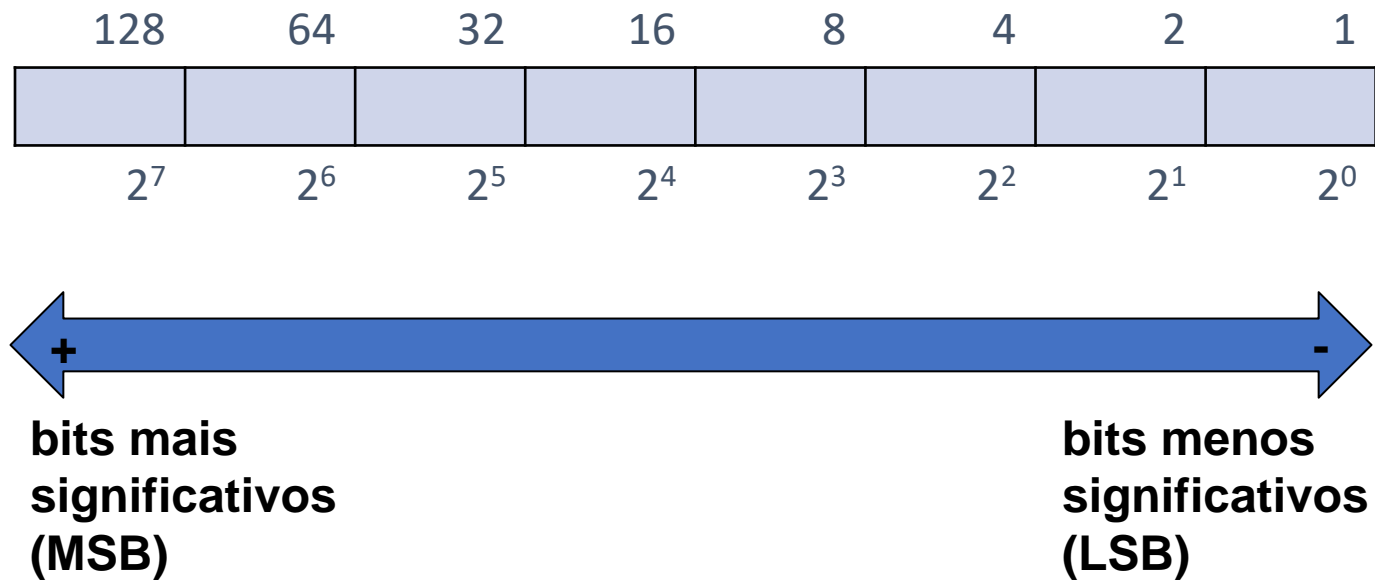
Cada dígito binário é chamado bit

Característica de valor posicional (casa)
cada posição vale o dobro da anterior:

Casa dos 1s, casa dos 2s, casa dos 4s, ...

Sistema Binário - Base 2

Posições:



Sistema Binário - Base 2

Exemplo:

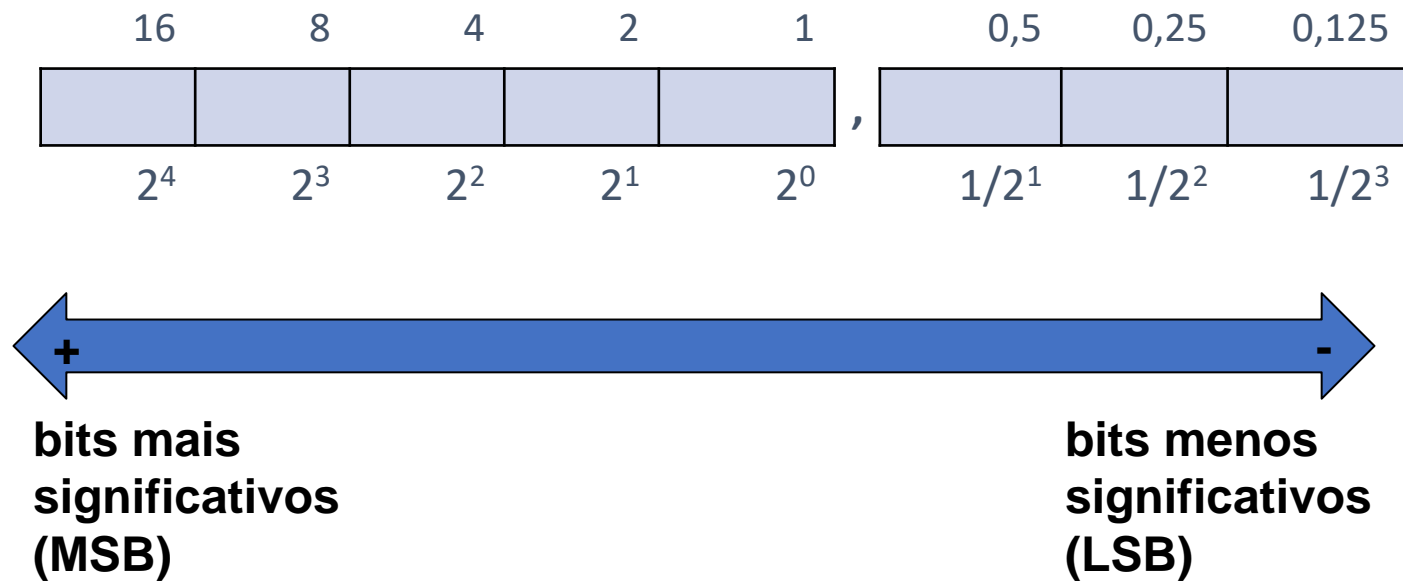
128s	64s	32s	16s	8s	4s	2s	1s
0	0	0	1	0	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

O número “zero, zero, zero, um, zero, zero, um, um” binário é obtido:
 $16 + 2 + 1 = 19$

$$10011_2 = 19_{10}$$

Sistema Binário - Base 2

Fracionários:



Sistema Binário - Base 2

Exemplo:

16	8	4	2	1		0,5	0,25	0,125
0	1	1	1	0	,	1	0	1
2^4	2^3	2^2	2^1	2^0		$1/2^1$	$1/2^2$	$1/2^3$

O número “zero, um, um, um, zero, vírgula, um, zero, um” binário é obtido:
 $8 + 4 + 2 + 0,5 + 0,125 = 14,625$

$$1110,101_2 = 14,625_{10}$$

Sistema Octal - Base 8

Uso de 8 símbolos para a representação de quantidades



Validade dos conceitos de *peso* e *posição*

Posições *não* recebem denominações específicas (como no sistema decimal)

Exemplo – 673_8 (Lê-se *seis sete três*)

Sistema Octal - Base 8

Exemplos

75310246_8 (Lê-se sete cinco três um zero dois quatro seis)

7	5	3	1	0	2	4	6
$7 \cdot 8^7$	$5 \cdot 8^6$	$3 \cdot 8^5$	$1 \cdot 8^4$	$0 \cdot 8^3$	$2 \cdot 8^2$	$4 \cdot 8^1$	$6 \cdot 8^0$

34717_8 (Lê-se três quatro sete um sete)

3	4	7	1	7
9408	1792	392	8	7

Sistema Hexadecimal - Base 16

Uso de *10* símbolos numéricos e *6* alfabéticos para a representação de quantidades

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Validade dos conceitos de *peso* e *posição*

Posições *não* recebem denominações específicas (como no sistema decimal)

Exemplo – $9FC_{16}$ (Lê-se *nove efe ce*)

Sistema Hexadecimal - Base 16

Exemplos

$7B3D_{16}$ (Lê-se sete be três de)

$$\begin{array}{cccc} \mathbf{7} & \mathbf{B} & \mathbf{3} & \mathbf{D} \\ \mathbf{7 \times 16^3} & | & \mathbf{11 \times 16^2} & | & \mathbf{3 \times 16^1} & | & \mathbf{13 \times 16^0} \end{array}$$

$FFA0_{16}$ (Lê-se efe efe a zero)

$$\begin{array}{cccc} \mathbf{F} & \mathbf{F} & \mathbf{A} & \mathbf{0} \\ \mathbf{15 \times 16^3} & | & \mathbf{15 \times 16^2} & | & \mathbf{10 \times 16^1} & | & \mathbf{0 \times 16^0} \end{array}$$

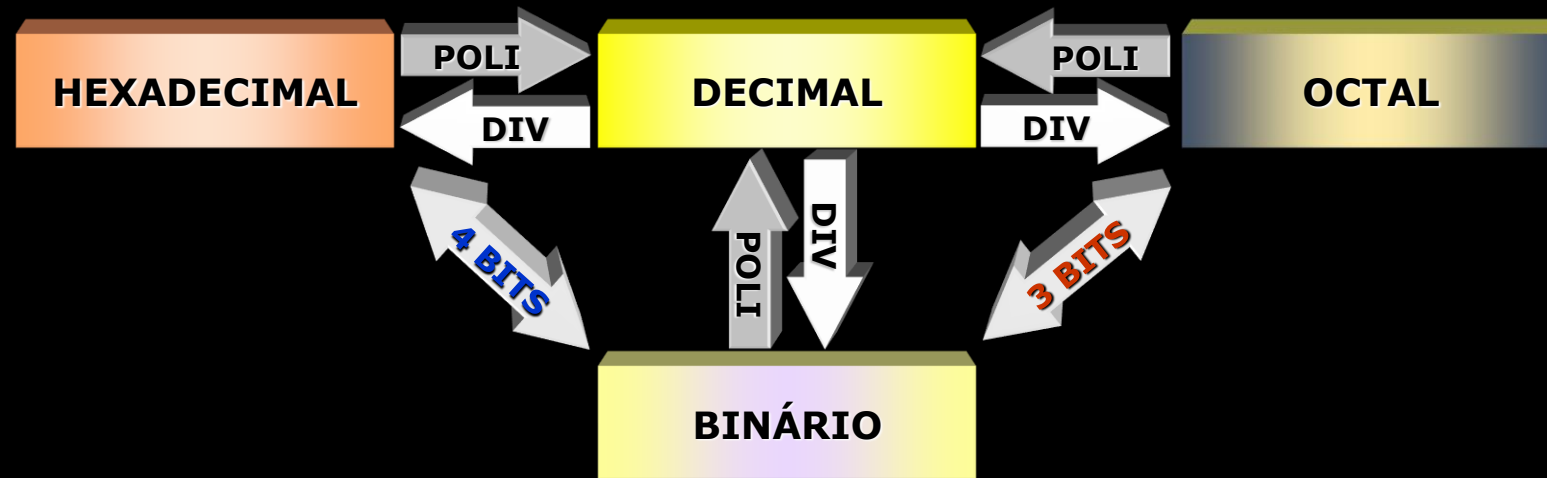
Conversão entre Sistemas de Numeração

Procedimentos Básicos para Números Inteiros

Divisão

Polinômio

Agrupamento de Bits



Conversão DECIMAL => BINÁRIO

Trabalha com divisão inteira + resto

$$87_{10} = 1010111_2$$

$$87 / 2 = 43 \text{ resto } 1$$

$$43 / 2 = 21 \text{ resto } 1$$

$$21 / 2 = 10 \text{ resto } 1$$

$$10 / 2 = 5 \text{ resto } 0$$

$$5 / 2 = 2 \text{ resto } 1$$

$$2 / 2 = 1 \text{ resto } 0$$

$$1 / 2 = 0 \text{ resto } 1$$

Conversão DECIMAL => BINÁRIO

Trabalha com divisão inteira + resto

$$87_{10} = 1010111_2$$

$$87 / 2 = 43 \text{ resto } 1$$

$$43 / 2 = 21 \text{ resto } 1$$

$$21 / 2 = 10 \text{ resto } 1$$

$$10 / 2 = 5 \text{ resto } 0$$

$$5 / 2 = 2 \text{ resto } 1$$

$$2 / 2 = 1 \text{ resto } 0$$

$$1 / 2 = 0 \text{ resto } 1$$

Verificação

64	32	16	8	4	2	1
1	0	1	0	1	1	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$64 + 16 + 4 + 2 + 1 = 87$$

Conversão DECIMAL => BINÁRIO

Conversão de decimal para binário

Nº 23

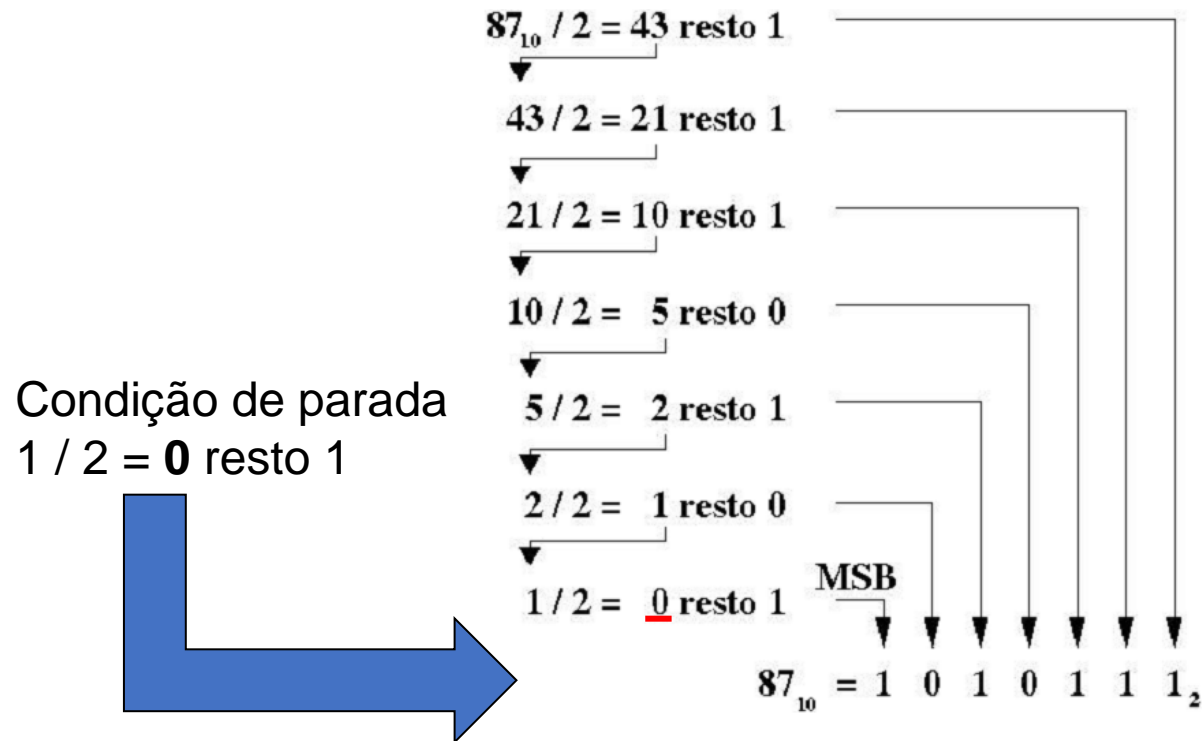
$$\begin{array}{r|l} 23 & 2 \\ \hline 1 & 11 \\ \hline \end{array} \longrightarrow 23 = 2 \times 11 + 1$$
$$\begin{array}{r|l} 11 & 2 \\ \hline 1 & 5 \\ \hline \end{array} \longrightarrow 23 = 2 \times (2 \times 5 + 1) + 1 = 5 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$\begin{array}{r|l} 5 & 2 \\ \hline 1 & 2 \\ \hline \end{array} \longrightarrow 23 = (2 \times 2 + 1) \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$\begin{array}{r|l} 2 & 2 \\ \hline 0 & 1 \\ \hline \end{array} \longrightarrow 23 = (1 \times 2) \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 23_{10}$$

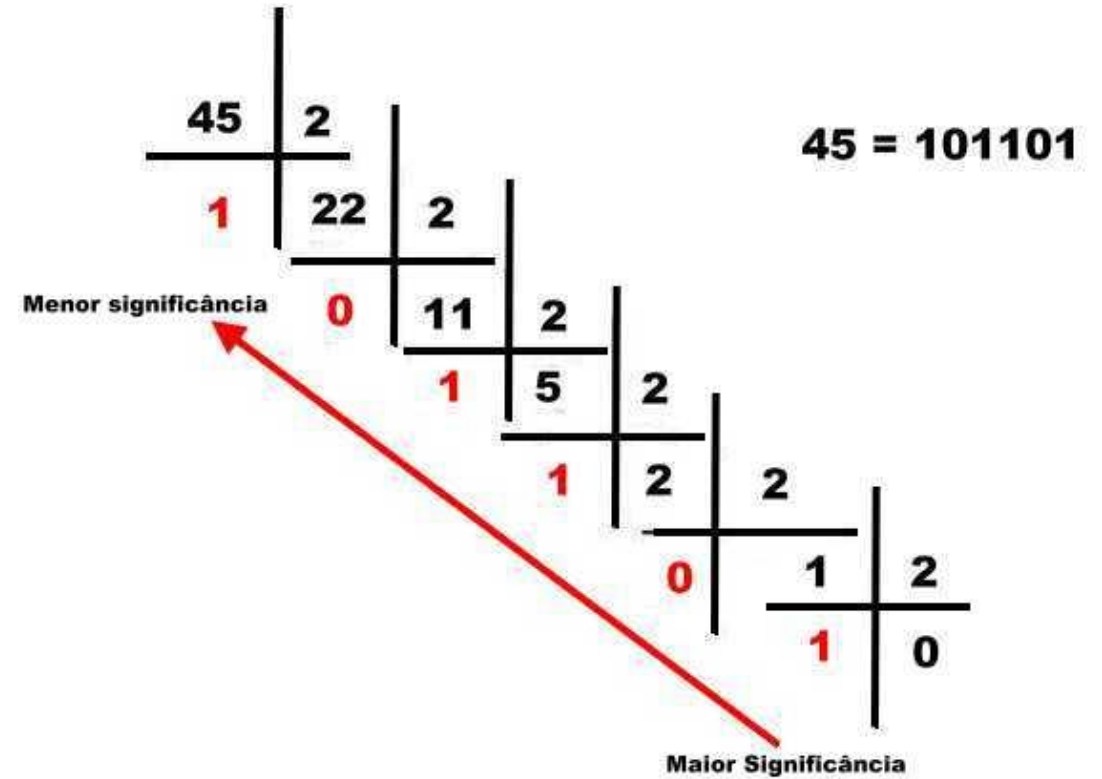
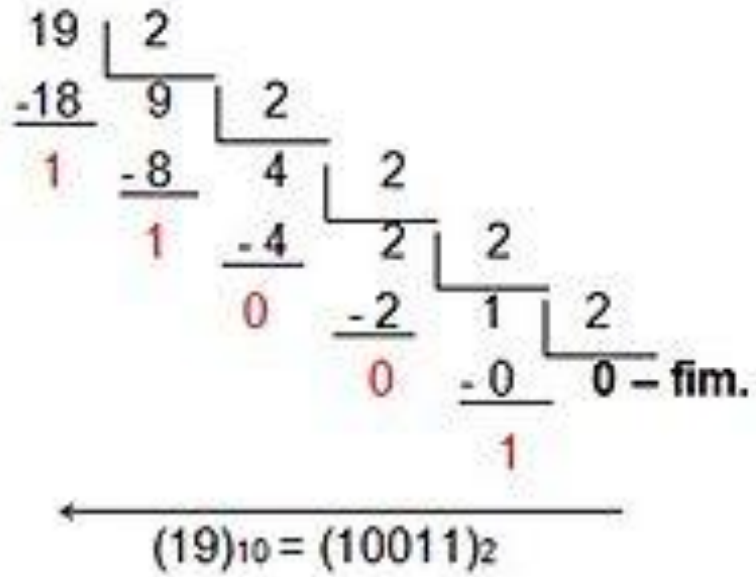
Regra Prática:

$$\begin{array}{r} 23 \\ \textcircled{1} \quad \text{11} \quad \text{5} \quad \text{2} \quad \text{1} \quad \text{0} \quad \text{1} \\ \hline \end{array}$$
$$\boxed{10111}_2 = 23_{10}$$

Conversão DECIMAL => BINÁRIO



Conversão DECIMAL => BINÁRIO



Conversão DECIMAL fracionário => BINÁRIO

A conversão da parte fracionária segue a seguinte regra prática:

Multiplica-se a parte fracionária pelo valor da base;

O número resultante a esquerda da vírgula é o dígito (0 ou 1) procurado;

Se o dígito à esquerda for 0 (zero) continuar a multiplicação pela base;

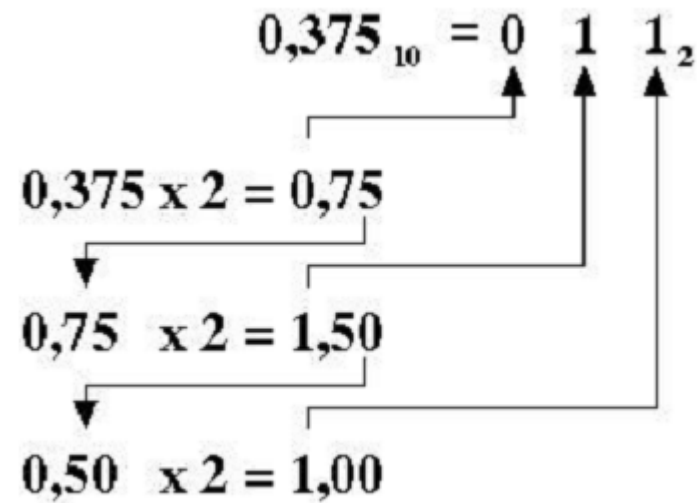
Se o dígito à esquerda for 1 este é retirado e prossegue-se a multiplicação;

O processo continua até obter-se 0 (zero) como resultado ou atingir-se a resolução estabelecida, no caso de dízima;

A leitura dos dígitos, ao contrário do caso da parte inteira, é feita de cima para baixo.

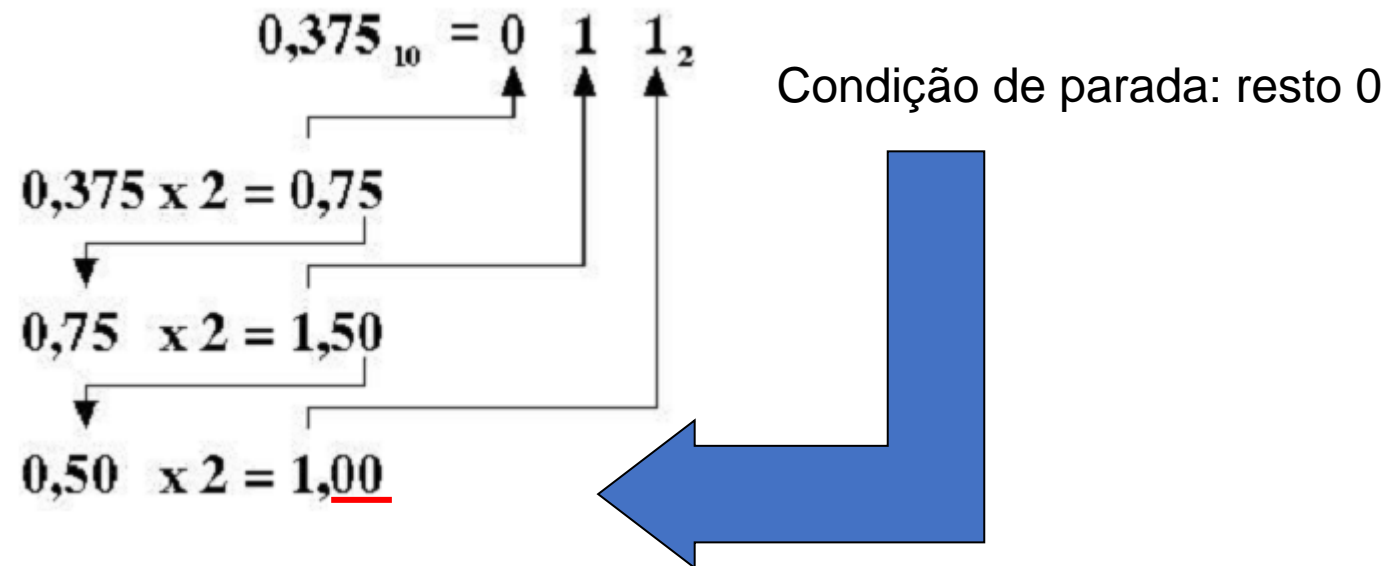
Conversão DECIMAL fracionário => BINÁRIO

$$0,375_{10} = 0,0\ 1\ 1_2$$



Conversão DECIMAL fracionário => BINÁRIO

$$0,375_{10} = 0,011_2$$



Conversão DECIMAL fracionário => BINÁRIO

$$35,625_{10} = ?_2$$

$$35,625_{10} = 35_{10} + 0,625_{10}$$

$\underbrace{\hspace{1.5cm}}_{\text{parte inteira}} \quad \underbrace{\hspace{1.5cm}}_{\text{parte fracionária}}$

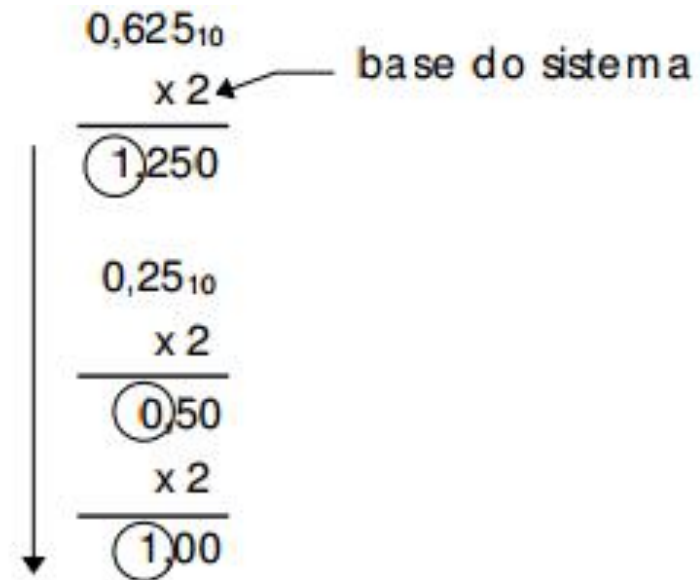
A conversão da parte inteira segue o procedimento já descrito:

$$35_{10} = 100011_2$$

Conversão DECIMAL fracionário => BINÁRIO

$$\begin{array}{r} 0,625_{10} \\ \times 2 \\ \hline \textcircled{1},250 \\ \\ 0,25_{10} \\ \times 2 \\ \hline \textcircled{0},50 \\ \\ \times 2 \\ \hline \textcircled{1},00 \end{array}$$

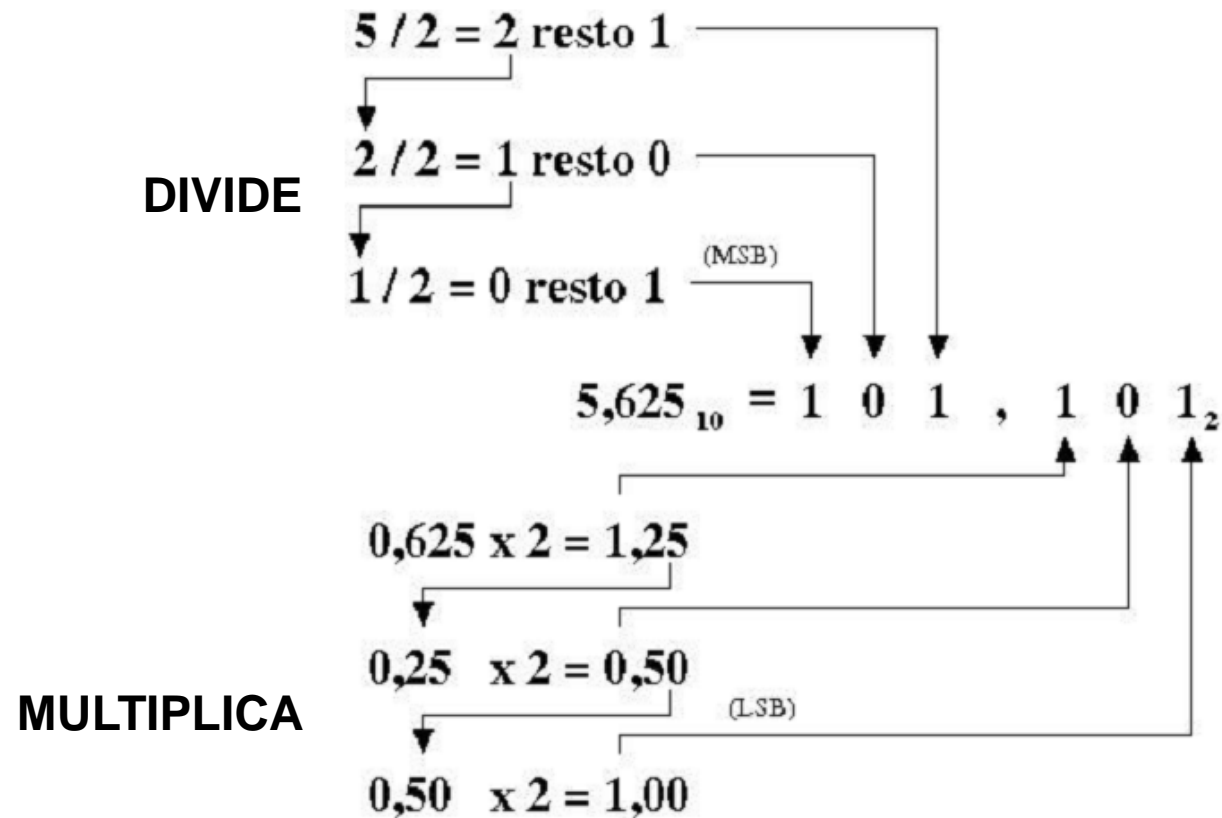
base do sistema



$$0,625_{10} = 0,101_2$$

$$35,625_{10} = 100011,101_2$$

Conversão DECIMAL fracionário => BINÁRIO



Conversão DECIMAL fracionário => BINÁRIO

$$\begin{array}{r} 8,7 \longrightarrow 8 \overline{) 2} \\ \underline{0} 4 \overline{) 2} \\ \underline{0} 2 \overline{) 2} \\ \underline{0} 1 \end{array}$$

1 0 0 0 , ? ? ?

$$2 \times 0,7 = 1,4 \longrightarrow 1$$

$$2 \times 0,4 = 0,8 \longrightarrow 0$$

$$2 \times 0,8 = 1,6 \longrightarrow 1$$

$$2 \times 0,6 = 1,2 \longrightarrow 1$$

1 0 0 0 , 1 0 1 1

Conversão BINÁRIO => DECIMAL

Multiplica-se cada dígito pelo valor da base elevada a uma dada potência, definida pela posição do dígito, e finalmente realiza-se a soma.

$$\begin{aligned}\text{Ex.: } 1001101_2 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 77_{10}\end{aligned}$$

Conversão BINÁRIO => DECIMAL

$$\begin{aligned}\text{Ex.: } 11001101_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 0 + 0 + 8 + 4 + 0 + 1 \\ &= 205_{10}\end{aligned}$$

Conversão BINÁRIO => DECIMAL


$$\begin{aligned} 1101,111_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 4 + 0 + 1 + 0,5 + 0,25 + 0,125 \\ &= 13 + 0,875 \\ &= 13,875_{10} \end{aligned}$$

Conversão de números fracionários


Conversão de números fracionários

Regra de Formação

$$\text{Decimal: } 197,526_{10} = 1 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 + 5 \times 10^{-1} + 2 \times 10^{-2} + 6 \times 10^{-3}$$



$$\text{Binário: } 101101,101 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$



Sistema OCTAL

A base de um sistema numérico é igual o número de dígitos que ela usa. Portanto, o sistema octal, que apresenta base 8, tem 8 dígitos a saber:

0, 1, 2, 3, 4, 5, 6, 7 (base $N = 8 \rightarrow$ dígitos $0 \rightarrow N-1 = 7$).

Sua utilidade nos sistemas digitais vem do fato de que, associando-se os algarismos de um número binário (bits) em grupos de três, obtém-se uma correspondência direta com os dígitos do sistema octal.

Conversão OCTAL => DECIMAL

Conversão de octal em decimal

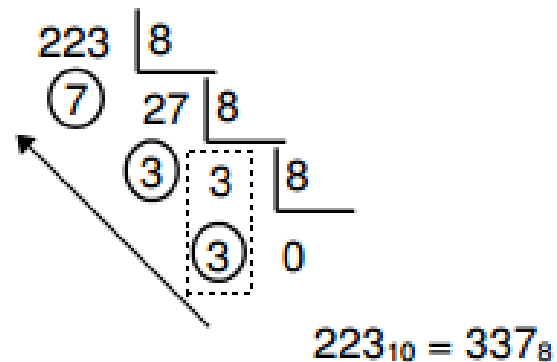
$$1247,235_8 = ?_{10}$$

$$\begin{aligned} &1 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3} \\ &512 + 128 + 32 + 7 + 2/8 + 3/64 + 5/512 \end{aligned}$$

$$1247,235_8 \sim 679,306_{10}$$

Conversão de decimal em octal

Nº 223

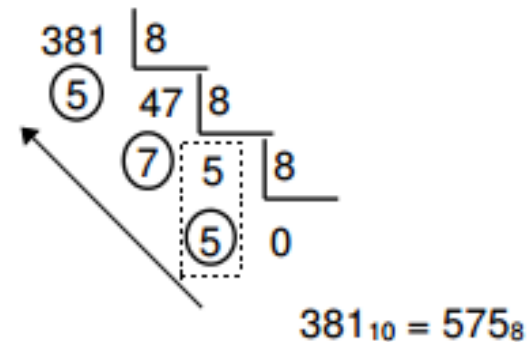


Conversão DECIMAL fracionário => OCTAL

Converter o número fracionário 381,796 da base decimal para octal (4 casas decimais após a vírgula).

$$381,796_{10} = 381_{10} + 0,796_{10}$$

Parte inteira:



Conversão DECIMAL fracionário => OCTAL

Converter o número fracionário 381,796 da base decimal para octal (4 casas decimais após a vírgula).

Parte fracionária:

$$\begin{array}{r} 0,796_{10} \\ \times 8 \\ \hline \textcircled{6},368 \\ 0,368_{10} \\ \times 8 \\ \hline \textcircled{2},944 \\ \times 8 \\ \hline \textcircled{7},952 \\ \times 8 \\ \hline \textcircled{4},416 \end{array}$$

$0,796_{10} \approx 0,6274_8$ (aproximado)

Conversão OCTAL => BINÁRIO

Para converter um número expresso em uma determinada base é normal convertermos o primeiro para um número na base 10 e, em seguida, fazer a conversão para a base desejada.

No caso do octal para o binário (e vice-versa) podemos fazer a conversão diretamente, sem passar pelo sistema decimal, já que, 8 é terceira potência de 2 e, portanto, são múltiplos e tem correspondência direta um com o outro.

Regra:

Cada dígito octal, a partir da vírgula, é representado pelo equivalente a três dígitos binários.

Conversão OCTAL => BINÁRIO

Tabela de equivalência

<i>Octa l</i>	<i>Binário</i>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



$$175,432_8 = 001\ 111\ 101,100\ 011\ 010_2$$

Conversão BINÁRIO => OCTAL

Converter binário em octal

Agrega-se os dígitos binários, a partir da vírgula, em grupos de três e converte-se para o equivalente em octal.

Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de três, adiciona-se zeros até que isto ocorra.

$$101110,011101_2 = \underbrace{101}_5 \underbrace{110}_6 , \underbrace{011}_3 \underbrace{101}_5_8$$

$$1011,11101_2 = \underbrace{001}_1 \underbrace{011}_3 , \underbrace{111}_7 \underbrace{010}_2_8$$

Converter o número 677_{10} para binário.

1ª alternativa: dividir 677_{10} sucessivamente por 2. Solução bastante extensa.

2ª alternativa: converter 677_{10} para octal e, em seguida, converter para binário.
(Solução menos trabalhosa).

$$677_{10} = 1245_8 = 1010100101_2$$

Sistema HEXADECIMAL

Este sistema apresenta base igual a 16.

Portanto 16 dígitos distintos.

São usados os dígitos:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Como no sistema de numeração octal, o hexadecimal apresenta equivalência direta entre seus dígitos e grupos de quatro dígitos binários.

Tabela

HEXADECIMAL	DECIMAL	OCTAL	BINÁRIO			
0	0	0	0	0	0	0
1	1	1	0	0	0	1
2	2	2	0	0	1	0
3	3	3	0	0	1	1
4	4	4	0	1	0	0
5	5	5	0	1	0	1
6	6	6	0	1	1	0
7	7	7	0	1	1	1
8	8	10	1	0	0	0
9	9	11	1	0	0	1
A	10	12	1	0	1	0
B	11	13	1	0	1	1
C	12	14	1	1	0	0
D	13	15	1	1	0	1
E	14	16	1	1	1	0
F	15	17	1	1	1	1

Conversão HEXADECIMAL => DECIMAL

A regra é a mesma da conversão de qualquer sistema de numeração para o decimal.

$$\text{AFC0,7D}_{16} = ?_{10}$$

$$A \times 16^3 + F \times 16^2 + C \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + D \times 16^{-2}$$

$$10 \times 16^3 + 15 \times 16^2 + 12 \times 16^1 + 0 \times 16^0 + 7 \times 16^{-1} + 13 \times 16^{-2}$$

$$44992,48828_{10}$$

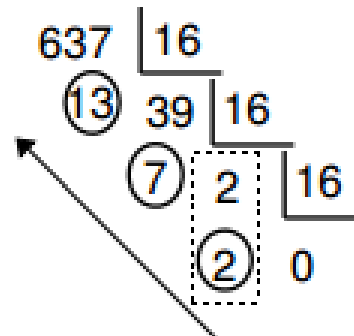
Conversão DECIMAL => HEXADECIMAL

A regra é a mesma da conversão do decimal para qualquer sistema de numeração

$$637,33_{10} = ?_{16}$$

$$637,33_{10} = 637_{10} + 0,33_{10}$$

Parte inteira



$$637_{10} = 27D_{16}$$

Parte Fracionária

$$\begin{array}{r} 0,33_{10} \\ \times 16 \\ \hline \textcircled{5}28 \end{array}$$

$$\begin{array}{r} 0,28_{10} \\ \times 16 \\ \hline \textcircled{4}48 \end{array}$$

$$\begin{array}{r} \times 16 \\ \hline \textcircled{7}68 \end{array}$$

$$\begin{array}{r} \times 16 \\ \hline \textcircled{10}88 \end{array}$$

$$0,33_{10} \approx 0,547A_{16} \text{ (aproximado)}$$

Conversão HEXADECIMAL => BINÁRIO

Da mesma forma que no sistema octal, não é necessário converter o número para o sistema decimal e depois para binário.

Basta representar cada dígito hexadecimal, a partir da vírgula, em grupos de quatro dígitos binários equivalentes.

A base 16 é a quarta potência da base 2.

A tabela de equivalência é a que foi apresentada mais acima.

$$FACA, CACA_{16} = ?_2$$

$$\begin{array}{ccccccc} F & A & C & A & , & C & A & C & A & _{16} \\ 1111 & 1010 & 1100 & 1010 & , & 1100 & 1010 & 1100 & 1010 & _2 \end{array}$$

$$FACA, CACA_{16} = 1111101011001010,1100101011001010_2$$

Conversão BINÁRIO => HEXADECIMAL

Como no caso da conversão de binário para octal, agrega-se os dígitos binários, a partir da vírgula, em grupos de quatro e converte-se para o equivalente em hexadecimal.

Caso os dígitos extremos, da direita ou esquerda, não formarem um grupo completo de quatro, adiciona-se zeros até que isto ocorra.

$$\begin{array}{r} 100101010,00111_2 = ?_{16} \\ 0001 \ 0010 \ 1010,0011 \ 1000_2 \\ 1 \quad 2 \quad A \quad , \quad 3 \quad 8_{16} \end{array}$$

$$100101010,00111_2 = 12A,38_{16}$$

ASCII

American Standard Code for Information Interchange

(Código Padrão Americano para o Intercâmbio de Informação)

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

Endereço IP – BINÁRIO => DECIMAL



Veja na figura os passos para converter um endereço binário para um endereço decimal.

No exemplo, o número binário:

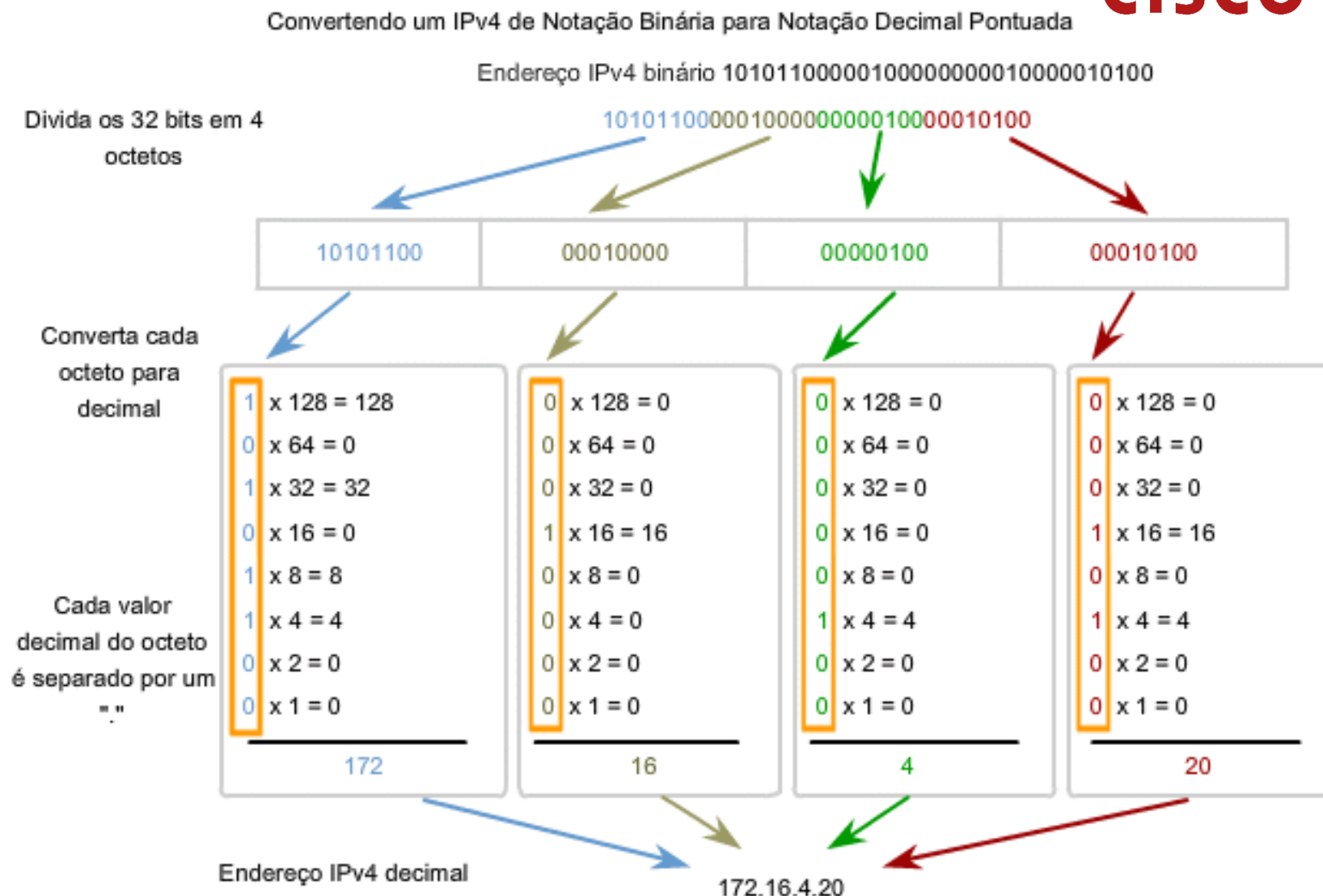
10101100000100000000010000010100

é convertido para:

172.16.4.20

Tenha em mente estes passos:

- Divida os 32 bits em 4 octetos.
- Converta cada octeto para decimal.
- Acrescente um "ponto" entre cada decimal







Não contavam
com minha
astúcia!

Exercícios

